

CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of tracking ~~reallocated~~ object data ~~when the object is moved for~~ during memory reclamation, comprising:
 - asserting ~~[[the]]~~ a status signal to indicate that the memory reclamation ~~reclaim~~ has begun on a memory block;
 - storing a physical address of a source memory location from where the object data is being moved;
 - storing a physical address of a destination memory location;
 - modifying the object data; and
 - moving the modified object data to the destination memory location.
2. (Currently Amended) The method of claim 1, further comprising ~~setting~~ deasserting the status signal to indicate the object data modification is complete.
3. (Currently Amended) The method of claim 1, wherein the object data is stored in flash memory.
4. (Original) The method of claim 3, wherein the flash memory is a flash electrically erasable read only memory (EEPROM) array.
5. (Original) The method of claim 1, wherein the status signal is a four (4) bit signal.
6. (Original) The method of claim 1, wherein the object data is an application program file.
7. (Original) The method of claim 6, wherein the application program file is a K-Java file.

8. (Currently Amended) An article comprising a machine readable medium having a plurality of machine readable instructions, wherein when the instructions are executed by a processor, the instructions cause a system to:

assert [[the]]a status signal to indicate that reclaim has begun on a memory block including an object;

store a physical address of a source memory location from where the object is being moved;

store a physical address of a destination memory location;

modify the object~~[[data]]~~; and

move the modified object~~[[data]]~~ to the destination memory location.

9. (Currently Amended) The article of claim 8 comprising a machine readable medium having a plurality of machine readable instructions, wherein when the instructions are executed by a processor, the instructions further cause [[a]]the system to:

~~[[set]]de-assert~~ the status signal to indicate the object~~[[data]]~~ modification is complete.

10. (Currently Amended) A method of modifying an object after a reclaim process, comprising:

reading [[an]]the object to be updated into a random access memory (RAM) location;

reading an input argument object size;

signaling an application for modification;

storing [[the]]a delta value by which [[the]] a pointer[[s]] within the object needs to be incremented; and

modifying the object ~~[[data]]~~if modification needs to be done.

11. (Currently Amended) The method of claim 10, wherein reading the object to be updated into the RAM location comprises reading the object currently stored at a

destination address within flash memory into the RAM location. ~~the objects are read to be updated from the destination address.~~

12. (Currently Amended) The method of claim 11, wherein modifying the object if modification needs to be done comprises:

updating the pointer within the object read into the RAM location based on the delta value; and

copying the modified object read into the RAM location back into the destination address within the flash memory.

~~wherein the object is updated in the destination address with the data passed into the allocated RAM space.~~

13. (Currently Amended) The method of claim ~~[[10]]~~12, wherein the application calls a function to modify the ~~[[data]]~~object ~~during reclamation.~~

14. (Currently Amended) The method of claim 13, wherein the function returns a Boolean value to indicate that the object modification is not necessary if the application does not need to modify the ~~[[data]]~~object.

15. (Currently Amended) The method of claim ~~[[10]]~~13, wherein the function allocates ~~[[the]]~~ RAM space required to ~~modify the data~~ read the object into the RAM location.

16. (Original) The method of claim 15, wherein the function allocates the RAM space specified by the input argument object size.

17. (Currently Amended) The method of claim 10, wherein the modification to the object ~~[[data]]~~is based on the delta value by which the pointer~~[[s]]~~ within the object needs to be incremented.

18. (Currently Amended) An apparatus, comprising:

a unique identifier to validate an object ~~[[data]]~~ in the apparatus;
a ~~plurality of~~ size field~~[[s]]~~ to indicate ~~[[the]]~~ a size of ~~[[each]]~~ the object to be moved during reclaim;
a ~~plurality of~~ status field~~[[s]]~~ to indicate ~~[[the]]~~ a status of ~~[[each]]~~ the object;
a source address field to track ~~[[the]]~~ a first memory location from where the object is to be moved; and
a destination address field to track ~~[[the]]~~ a second memory location to where the object is to be moved.

19. (Original) The apparatus in claim 18, wherein the apparatus is a paragraph object.

20. (Currently Amended) The apparatus of claim 18, wherein the apparatus can store an unlimited number of objects. ~~[[data.]]~~

21. (Currently Amended) The apparatus of claim 20, wherein each object ~~[[data]]~~ has a corresponding status field and size field.

22. (Currently Amended) A method of reclaiming a flash memory block, comprising:

setting a flag to prevent a file interpreter from reclaiming another memory block at the same time;

requesting permission from the file interpreter to do a reclaim on the memory block;

moving valid data to a spare memory block, the valid data being data not scheduled to be erased;

compressing the valid data;

erasing the memory block; and

copying the valid data back into the memory block from the spare memory block.

23. (Currently Amended) The method of claim 22, further comprising:

clearing the flag to indicate that the reclaim on the memory block is
complete[[d]].